Using the RAP System to Simulate Human Error

Michael Freed
NASA Ames Research Center
email: mfreed@mail.arc.nasa.gov

Introduction

Newly designed equipment and procedures often inadvertently facilitate human error. Techniques for identifying error facilitations in design tend to be either ineffective or very expensive. For example, one of the most effective ways to test new designs is to hire human operators to carry out tasks using prototyped equipment, and then observe their performance in a wide range of operating conditions. In our domain, air traffic control, such tests typically require hiring highly paid expert controllers as subjects, often for extended periods. The limited amount of testing that results from high cost stifles innovation and compromises safety.

One way to get some of the benefits of a "human in the loop" study at much lower cost is to use a computer to simulate all elements of such a study including the equipment, human operators, and experimental observers. Human simulation has been used successfully by others to guide design (e.g. [John90,Corker95]). However, operator performance models used in previous human simulation systems have been unable to carry out complex tasks such as air traffic control. Our system uses a powerful operator model that overstates human capabilities in some ways, but can operate in domains where predicting error would be most useful.

In order to build a human model able to function in interesting domains, we have adapted planning and control mechanisms developed by AI researchers. Air traffic control (ATC), like many of the domains in which the prediction of design-facilitated error would be most useful, requires the ability to interleave the execution of multiple tasks in complex, dynamic, and highly routinized task environments. These domain attributes made the RAP execution system [Firby89] an especially suitable starting point for the model

Though not initially designed to make errors of any kind, we have adapted the RAP system to help predict a type of error sometimes referred to as a "habit capture" [Reason90]. Habit captures are defined by their apparent cognitive cause. In particular, people make such errors when, instead of deliberating, they act on a false but usually reliable habit or assumption. Habit captures are reported quite frequently in naturalistic studies of error [Reason82, Norman88]. For example:

> "I went to the bedroom to change in to something more comfortable for the evening, and the next thing I knew I was getting into my pajama trousers, as if to go to bed. “
>
> "I had decided to cut down my sugar consumption and wanted to have my cornflakes without it. But the next morning, however, I sprinkled sugar on my cereal just as I always do."

To reproduce habit capture errors in simulation, our model attempts to predict how and when people use assumptions to select and specify action. In conjunction with mechanisms for simulating an ATC task environment, the model allows us to identify circumstances in which a person might rely on false assumptions. This paper describes an example of how our system employs plan execution mechanisms to simulate and predict design-facilitated habit-capture errors.

Human Model

The human operator model describes a set of cognitive, perceptual, and motor components, each of which can contribute to performance errors in some circumstances. Habit captures arise mainly within elements of the cognitive portion of the model that determine how certain resources are apportioned in deciding between alternatives. The cognitive model consists of five components.

This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.
File Name : f-hom.ps
Pages : 1

Overview of the Human Operator Model

The **interpretation** component produces potentially goal-relevant classifications of perceived objects and then encodes those classifications in **working memory**. For example, the detection of a nearby highway exit may be construed by interpretation mechanisms as an opportunity to get off the highway. When there are alternative interpretations for some perception, the **decision** component selects between them.

The **action** component coordinates physical and mental tasks in accordance with "sketchy" routine plans and a set of task interleaving heuristics [Firby89]. When there are alternative courses of action available, the decision component selects between them. For example, the decision component would select between possible driving routes and between alternative ways to pass the time while stuck in traffic.

Together, the action and decision components incorporate much of the functionality of the RAP execution system and thus enable the model to operate in complex, dynamic domains such as air traffic control. The decision component changes the way the RAP system decides between alternatives --- e.g. when selecting a method to achieve a goal. In particular, decision component mechanisms employ a flexible *decision strategy* specifying how scarce information acquisition resources such as visual attention and mechanisms for memory retrieval [Carrier95] will be used to decide between alternatives.

For example, the strategy for deciding whether to exit the highway at the usual place (Main Street) or continue on might involve combining information about one's current destination, whether to expect rush hour traffic, and whether to expect road construction delays. The *default decision strategy* prescribes particular methods for evaluating each of these factors --- e.g. *retrieving* one's current destination from memory, visually *scanning* the dashboard clock to determine the current time (whether to expect rush hour traffic) and *assuming* the absence of construction on Main St.

Transient *biases* stored in **bias memory** temporarily modify a default decision strategy. For example, hearing about construction on Main Street over the radio and encoding this fact in working memory results in bias that changes how the default strategy handles the road construction factor; whereas the default strategy prescribes relying on a default assumption, the bias-modified strategy retrieves road construction information from working memory. Biases are produced when items are encoded in working memory and last until a fixed expiration interval has passed. If the working memory item is reencoded or retrieved for any reason, the bias is refreshed --- i.e. the expiration interval is reset.

In routine operating conditions, the default decision strategy and current bias combine to produce an effective "current decision-making strategy.” In unusual conditions or in a poorly designed task environment, the current strategy may prove inadequate. In particular, the current strategy may prescribe relying on an inaccurate default assumption rather than checking the assumption against information from working memory or the immediate perceptual environment. Actions selected as a result of such decisions produce habit capture errors.

An Example

At a TRACON air traffic control facility, one controller will often be assigned to the task of guiding planes through a region of airspace called an approach sector. This task involves taking planes from various sector entry points and getting them lined up at a safe distance from one another on landing approach to a particular airport. Some airports have two parallel runways. In such cases, the controller will form planes up into two lines.

Occasionally, a controller will be told that one of the two runways is closed and that all planes on approach to land must be directed to the remaining open runway. A controller's ability to direct planes exclusively to the open runway depends on remembering that the other runway is closed. How does the controller remember this important fact? Normally, the diversion of all inbound planes to the open runway produces an easily perceived reminder. In particular, the controller will detect only a single line of planes on approach to the airport, even though two lines (one to each runway) would normally be expected.

However, problems may arise in conditions of low workload. With few planes around, there is no visually distinct line of planes to either runway. Thus, the usual situation in which both runways are available is perceptually indistinguishable from the case of a single closed runway. The lack of perceptual support would then force the controller to rely on memory alone, thus increasing the chance that the controller will accidentally direct a plane to the closed runway.

Simulation

When the simulated controller (sim) hears that the left runway is closed, interpretation mechanisms cause a propositional representation of this fact to be encoded in working memory. The encoding event generates bias according to the following rule:

```
Bias-rule
  Trigger-condition:(closed ?rwy)
  Do:(create-bias decision-procedure-
27    step5 :duration (10 minutes))
```

Newly generated bias is represented explicitly in priming memory along with a notation indicating when the bias will expire if not renewed.[1] In this case, bias lasting 10 minutes causes decision mechanisms to consider the possibility of runway closure (step5) in cases where the usual state --- all runways open --- might otherwise be assumed.

When a plane approaches its airspace, the simulated controller initiates a routine plane-handling procedure involving accepting responsibility for the plane, determining where the plane is headed, and then guiding it to its destination. If the plane's destination is Los Angeles airport (LAX), guiding it to its destination will involve selecting between the airport's two parallel runways.

For highly routine decisions such as runway selection, human controllers can reasonably be expected to know which factors to consider in making the decision and how to appropriately weight each factor. Our model reflects this by treating routine decision-making as a proceduralized task, not distinguished in any way from proceduralized physical action. Decision procedures consist of a set of steps for acquiring information about decision-relevant factors and then combining the information to make a choice.

```
Decision-procedure27: select ?rwy for
      ?plane
1)  determine which runway has fewer
    planes    on approach => ?factor1
2)  determine which runway approach
    would     be fastest for ?plane =>
    ?factor2
3)  determine which runway approach
    would     be easiest for me =>
    ?factor3
4)  determine which runway has better
      microclimate => ?factor4
5)  determine whether left runway is
    open      => ?factor5
6)  determine whether right runway is
    open      => ?factor6
7)  determine which runway currently
    safest    for  ?plane => ?factor7
8)  compute-decision:
    f(factor1,factor2..)
```

In most cases, there will be more than one method for acquiring information about a factor. In this example, the controller could determine the status of the left runway by retrieving information from memory, asking another controller, or by assuming the most likely condition --- that the runway is open. The *default method*, which in this case is to assume the most likely condition, will always

[1] We assume that human decision mechanisms gradually refine bias rule expiration intervals and that bias associated with routine decisions endure for an approximately optimal period (q.v. [Anderson93]): t=S*min(L,R) where S = a safety factor, L = the expected lifetime of the observed fact or intention, and R = expected rate at which bias will be replenished.

be selected unless bias in priming memory promotes some (more effortful) alternative.[2]

By default, the sim assumes that the left runway is available. Bias produced after learning of the runway closure causes the sim to temporarily override the default, and instead verify the runway's availability by retrieving information from memory. Runway closure information will for some time thereafter be retrieved from working memory whenever a runway selection occurs.

Eventually, the initial bias expires. To select a runway for a newly arrived plane, the sim will once again consider only the default assumption. Other factors will then determine which runway is selected. For example, the controller may choose to direct a heavy plane to the longer left runway which, in normal circumstances, would allow the plane an easier and safer landing. With the left runway closed, actions following from this decision result in error.

Avoiding error requires maintaining appropriate bias. In a variation of the described scenario in which no error occurs, visually perceived reminders of the runway closure cause bias to be periodically renewed.[3] In particular, whenever visual attention mechanisms attend to plane icons on an approach path to the airport, interpretation mechanisms note the absence of a line of planes to the left runway and signal an expectation failure.

```
Expectation-generation-rule
   Context: (and (> perceived-workload low) ..)
   Do: (create-task verify
         (on approach-path left (colinear-visual-group
                    plane-icons)))
```

In general, whenever an expectation failure occurs, the action component responds by initiating a task to explain the observed anomaly. The first step in such a task is to try to match the anomaly to a known explanation-pattern (XP) [Schank86]. If a match is found, action mechanisms then attempt to verify the hypothesis provided by the XP.

```
Explanation-pattern-15
   Anomoly: (not (on approach-path ?rwy  (colinear-
          visual-group plane-icons)))
   Candidate Explanation: (closed runway ?rwy)
   Verify-action:              (working-memory-retrieve
          (closed   runway ?rwy))
```

In principle, verifying a hypothesis could involve mental and physical actions of any kind. In the current model however, the only way to verify a hypothesis is to check for a match in working memory. In this case, the contents of working memory are adequate; the anomalous absence of planes on approach to the left runway is explained as a result of the left runway's closure.

Bias renewal occurs whenever the working memory item that originally produced the bias is reencoded or retrieved. Thus, retrieving the proposition (closed runway left) triggers the bias generation rule just as if the proposition had been encoded. Thus, the unusual arrangement of planes on the radar scope acts as a constant reminder, preventing the sim from reverting to the use of its default assumption and thereby preventing error.

Conclusion

One way to get some of the benefits of a "human in the loop" study at much lower cost is to use a computer to simulate all elements of such a study including the equipment, human operators, and experimental observers. Our approach uses a human operator model adapted from the

[2] The set of default methods for steps of a given decision procedure constitute that procedure’s *default decision strategy*.

[3] In the initial model, all visual objects are attended to at all times

RAP plan execution system to enable the simulated operator to function in domains where predicting error would be most useful.

The model can currently predict some circumstances in which humans would tend to make habit capture errors. The key model elements for these predictions regulate the use of scarce cognitive and perceptual resources in making decisions. When inadequate resources are allocated to a decision task, the model relies on fallible assumptions, thus rendering the simulated operator vulnerable to error. Habit captures occur quite often in everyday life and, arguably, in the air traffic control domain. By alerting designers to the potential for such errors early in the design process, we hope to reduce the cost of evaluation and thereby speed the safe introduction of new technology.

Acknowledgements

The research described in this paper has benefited from discussions with many talented researchers including Jim Johnston, Roger Remington, Eric Ruthruff, Mark Van Selst, and Mike Shafto.

References

[Anderson90] Anderson, J.R. The Adaptive Character of Thought. Lawrence Earlbaum Associates, Hillsdale, N.J., 1990.

[Carrier95] Carrier, L.M. and Pashler, H. Attentional limitations in memory retrieval. Journal fo experimental psychology: learning, memory, & cognition, 21, 1339-1348.

[Corker93] Corker, K.M. and Smith, B.R. An architecture and model for cognitive engineering simulation analysis. Proceedings of the AIAA Computing in Aerospace 9 Conference, San Diego, CA, 1993.

[Firby89] R.J. Firby. Adaptive execution in complex dynamic worlds. Ph.D. thesis, Yale University, 1989.

[John94] John, B.E. and Kieras, D.E. The GOMS Family of Analysis Techniques: Tools for Design and Evaluation. Carnegie Mellon University. School of Computer Science, TR CMU-CS-94-181, 1994.

[Norman88] Norman, Donald A. The Psychology of Everyday Things. Basic Books, New York, N.Y., 1988.

[Reason90] Reason, J.T. Human Error. Cambridge University Press, New York, N.Y., 1990.

[Reason82] Reason, J.T. and Mycielska, K. Absent-minded? The psychology of mental lapses and everyday errors. Englewood Cliffs, N.J., Prentice Hall, 1982.

[Remington90] Remington, R.W., Johnston, J.C., Bunzo, M.S., and Benjamin, K.A. The cognitive simulation system: an interactive graphic tool for modelling human cognitive reasoning. In Objected-oriented Simulation, San Diego: Society for Computer Simulation, pp. 155-166, 1990.

[Schank86] Schank, Roger C. Explanation Patterns. Lawrence Earlbaum Associates, Hillsdale, N.J., 1986.

[Stein93] Stein, Earl S. and Garland, Daniel. Air traffic controller working memory: considerations in air traffic control tactical operations. FAA technical report DOT/FAA/CT-TN93/37, 1993.